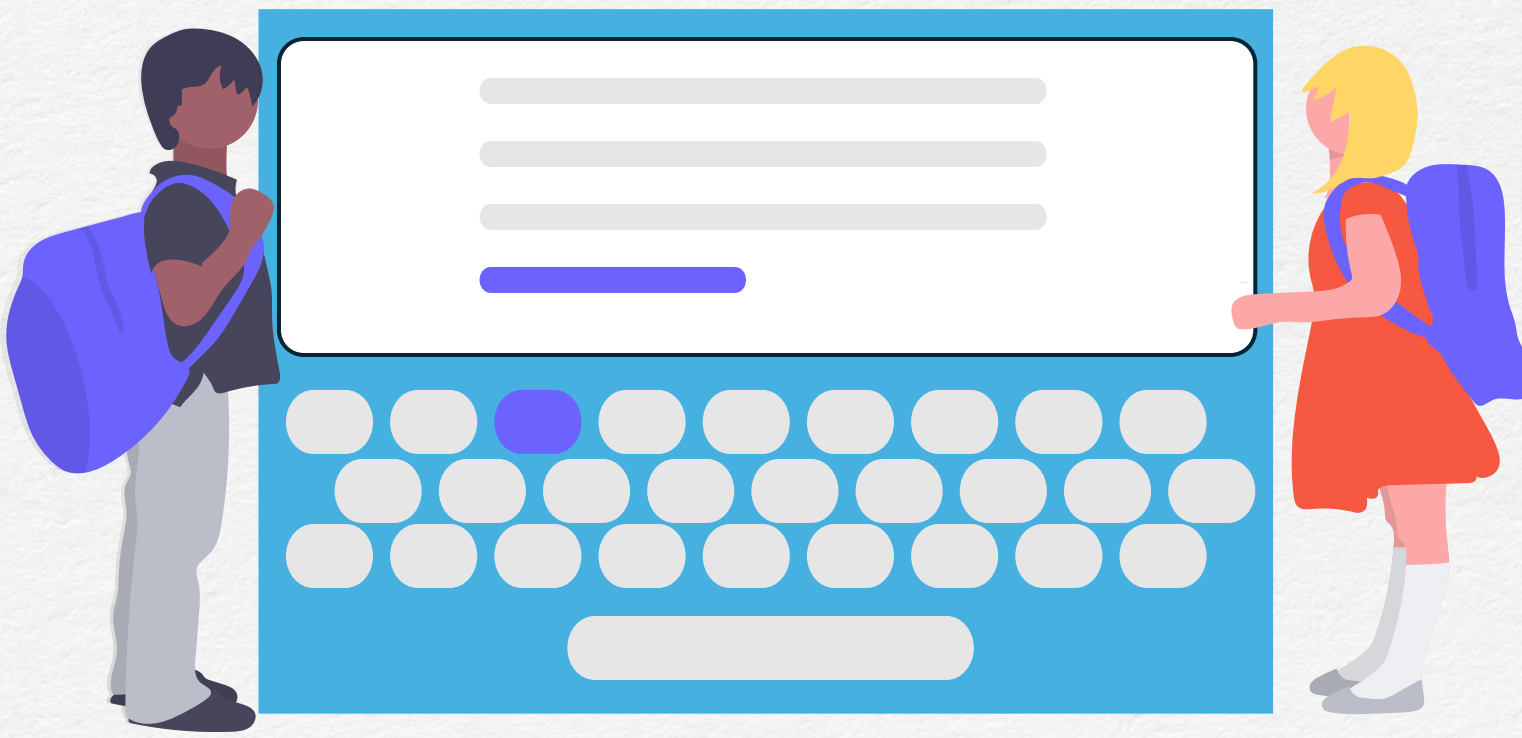


Physical Product Design:

# A Dedicated **Coding** **Computer** for Young **Learners**

original design by Marian Veteanu



# The “Coral Micro” Computer

an original design by Marian Veteanu

This presentation introduces the concept and design of a dedicated, single-purpose computer aimed at introducing young coders to the exciting world of programming.

The computer is designed as a STEM toy that offers real coding capabilities.



# Why?

In the 80s, microcomputers provided young coders with a quick path to programming by booting directly into a BASIC environment.

Although many efficient coding environments exist on modern computers, these devices also come with numerous distractions (games, video platforms, etc.) that compete for a young person's attention.

There are already a wide range of STEM devices on the market that claim to teach coding to young people. Although many of these devices are captivating, their educational value is often subpar, making them little more than simple toys.

The availability of low-cost SBCs (Single Board Computers) and SoC (System on Chip) technology has enabled manufacturers to add "smart" capabilities to products at all price points. As a result, creating a dedicated, low-cost coding computer for young coders is now more achievable than ever. Such a computer can provide a serious coding education through fun and engaging activities.

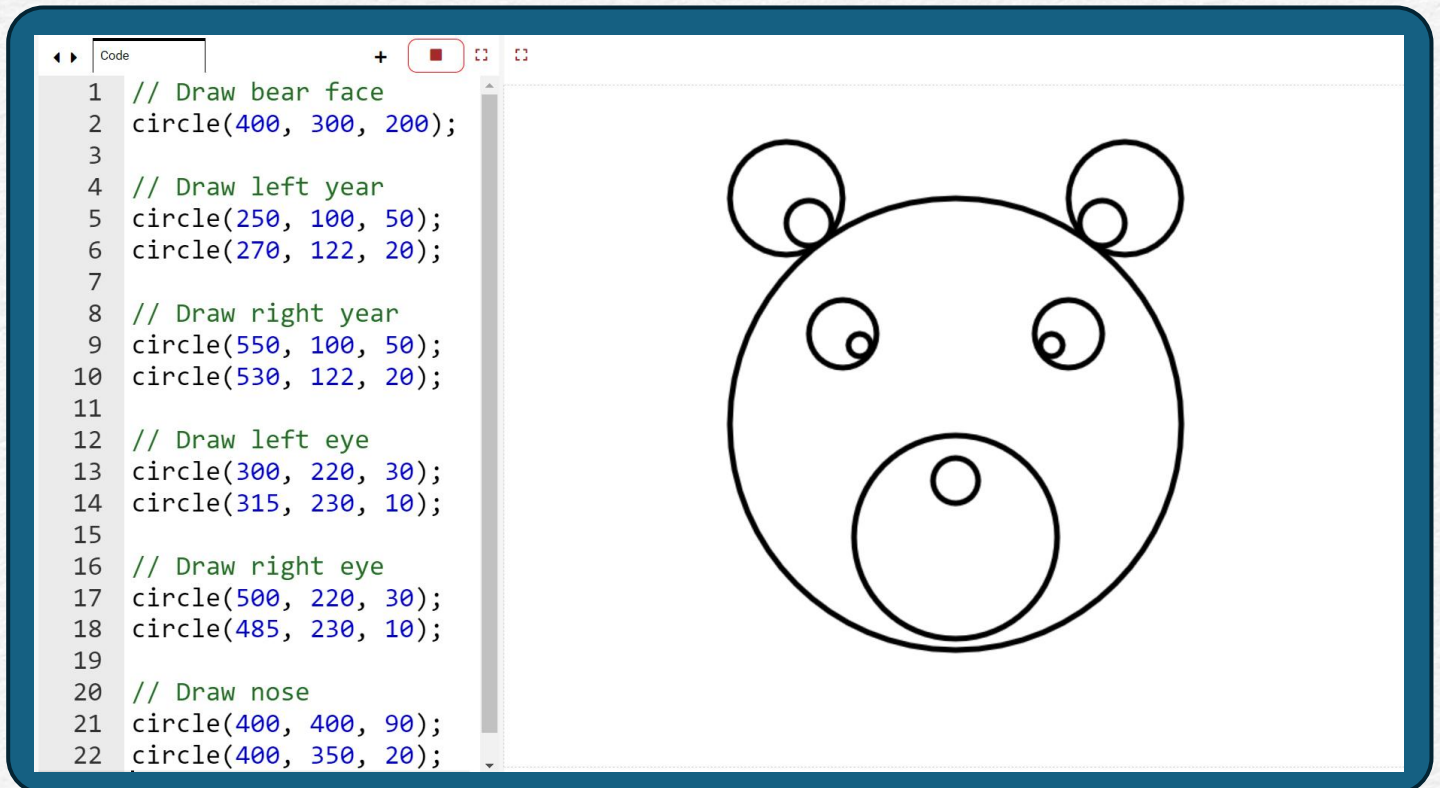
Introducing: **“Coral Micro”**  
(a dedicated coding computer for young learners)



# Requirements

- The computer should be marketed as a STEM toy and displayed on shelves of typical retailers in the STEM category.
- The "Coral Micro" coding computer should feature a distinctive design that is attractive and appeals to both kids and adults. The computer should have an AIO design so it can be used immediately as is taken out of the box, but also feature HDMI out for connection to an external monitor.
- The computer should come packaged with an illustrated, spiral-bound coding book that includes project-based activities (less text, more practice).
- The computer should boot in under 3 seconds into a simplified coding environment with a minimal set of features that allows users to:
  - Create new programs/apps
  - Load an existing app
  - Browse the public "app store" to discover new programs
  - Share programs with other users
  - Submit programs to the public "app store" (with parent approval)
- The computer should enable users to program in a simple text-based language, such as JavaScript, Python, or Lua.
- The computer should facilitate the development of apps that appeal to young coders (e.g., games).
- The computer should include a GPIO interface for physical computing.
- The computer should be powered by a low-power chip (e.g. ARM or RISV-V) and have an optional battery.

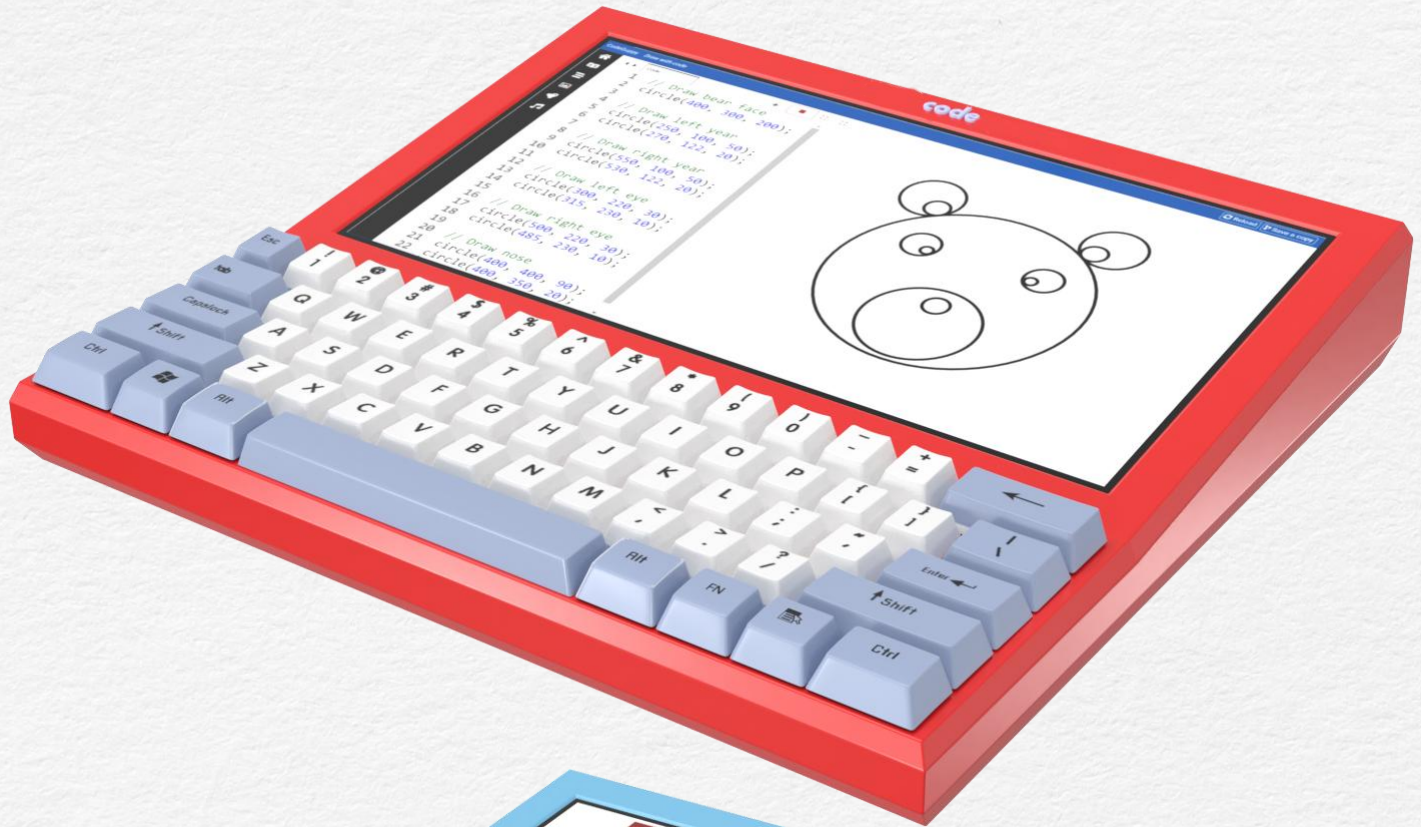
# Consideration for the Coding Experience



- The “Coral Micro” computer should offer a modern, easy to use coding environment that can be used just from the keyboard (no mouse)
- The environment should offer a “zero boiler-plate” coding experience to allow students learn concepts in a natural way
- The environment should offer graphical oriented functions such as drawing functions and for working with sprites.



# Computer mockups



To offer a differentiator factor, the computer can have a slate design and can come in multiple colors. The computer should feature a good quality keyboard optimized for typing.



# Book mockups

## Create a Breakout clone in JavaScript



Breakout is a classic arcade video game that is still enjoyable even today.

The original Breakout game was developed and published by Atari back in 1976. The game was an instant hit and inspired countless of clones on all generations of computers. Your parents probably played the original Breakout or one of the numerous clones on an 8 bit computer such as Commodore 64 or ZX-Spectrum 48K.

In this tutorial you'll learn how to recreate this game in JavaScript. We're pretty sure that this classic game will offer you and your friends many hours of entertainment.

The program has the typical arcade game structure, the main structure being the `loop()` function that gets executed 60 times per second.

```
var paddleWidth = 60;
var brickWidth = 50;
var brickHeight = 20;
var brickSpace = 20;
var rowSpace = 10;

var ball;
var ballsLeft;

var bricks = [];

background('Field');
fill("white");

function enter()
{
  initGame();
}

function initGame()
{
  bricks = createBricks();
  ballsLeft = 3;

  initBall();
}
```

143

```
t
if ( !ball.inMotion )
{
  ball.x = xPaddle + paddleWidth
/ 2;
  ball.y = yPaddle - ball.radius;
}
else
{
  updateBallInMotion();
}
}
```

```
t
clear();

readKeys();
displayBricks();
displayPaddle();

updateBall();
displayBall();

checkForCollision();

displayStats();
}
```

If you typed-in the program correctly, you should be able to see the game running when you press the *Play* button.

Feel free to customize the background of this game by selecting one of the many included background images. Your computer comes with many nice assets built-in, so you can focus on learning to code and not on finding images.



After you customize the game to your liking, you can share it with your friends or post it on your social media wall. Your friends won't need a *CORALMICRO* to play the game. Any modern browser on a regular computer will be enough to play it.

144

Fun JavaScript projects for  
your *CORALMICRO*



Learn to code on your *CORALMICRO* using type-in programs  
Build fun and engaging games using real coding  
Great alternative to block-based coding for early coding education

Adrian Veteșanu



- Dedicated coding computer
- Attractive form factor for both kids and adults
- HDMI output for connecting to external monitor
- WiFi (used only by app store and program sharing)
- USB ports for mouse (optional) and external storage
- GPIO ports for connecting sensors and actuators
- Powered by Raspberry Pi (without Linux experience)
- Hackable
- Built-in programming language JavaScript (or Lua, Python)
- Built-in APIs for graphics, sprites, sounds
- Boots to coding environment in 3s
- Dedicated “Run” and “Stop” program buttons on keyboard
- Come bundled with 300 pages spiral bound coding book
- App store for downloading new programs (in source code)
- Submit your program to app store or share it with friends
- Friends can run your program on any computer in a web browser

Available as bundle (COMPUTER + BOOK) in the STEM section of your favorite retailer





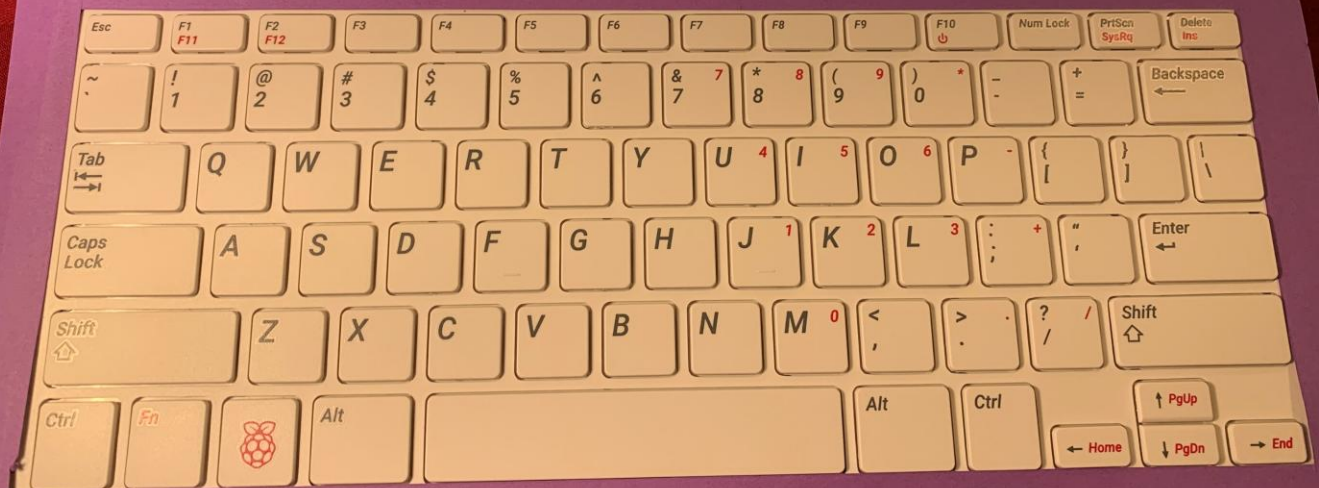
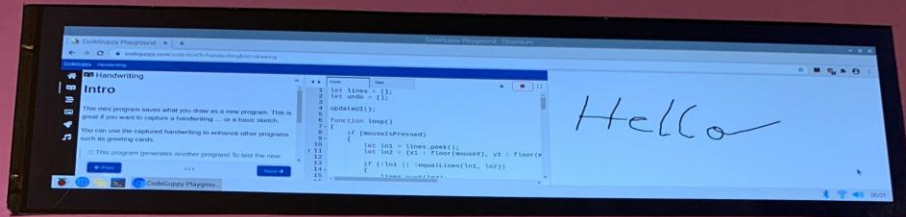


- CORALMICRO is inspired by the great 8-bit home computers from the golden-age of microcomputers
- Boots directly to coding environment
- Comes bundled with book with type-in programs
- Users learn to code by typing in programs from the book
- Encourages kids to read books
- Typing-in increases attention to details
- Real-world programming in JavaScript (or Lua, Python)
- Additional books and resources may become available from 3<sup>rd</sup> parties

# Prototypes

A few physical prototypes were executed to validate the form factor and user experience.

**Prototype 1:** A cardboard prototype built using a Raspberry Pi 400 computer and an ultra-wide LCD. The prototype showed that ultra-wide LCDs offer a distinctive appeal but make the computer harder to program.







**Prototype 2:** This prototype is designed around a keyboard with LCD and an embedded SBC (Raspberry Pi 5)



**Prototype 3:** This prototype is a custom design using a 3D printed enclosure. Inside there is an SBC computer (Raspberry Pi 5). The design played with a different aspect ratio LCD.

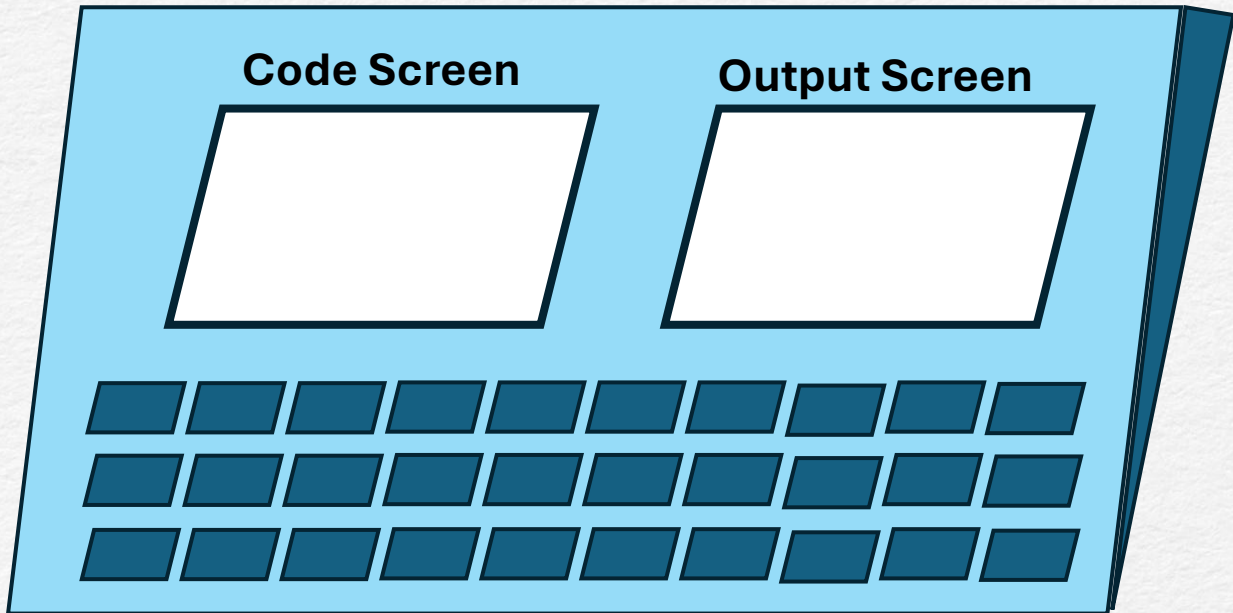
The “Coral Micro” logo can be seen on the right.



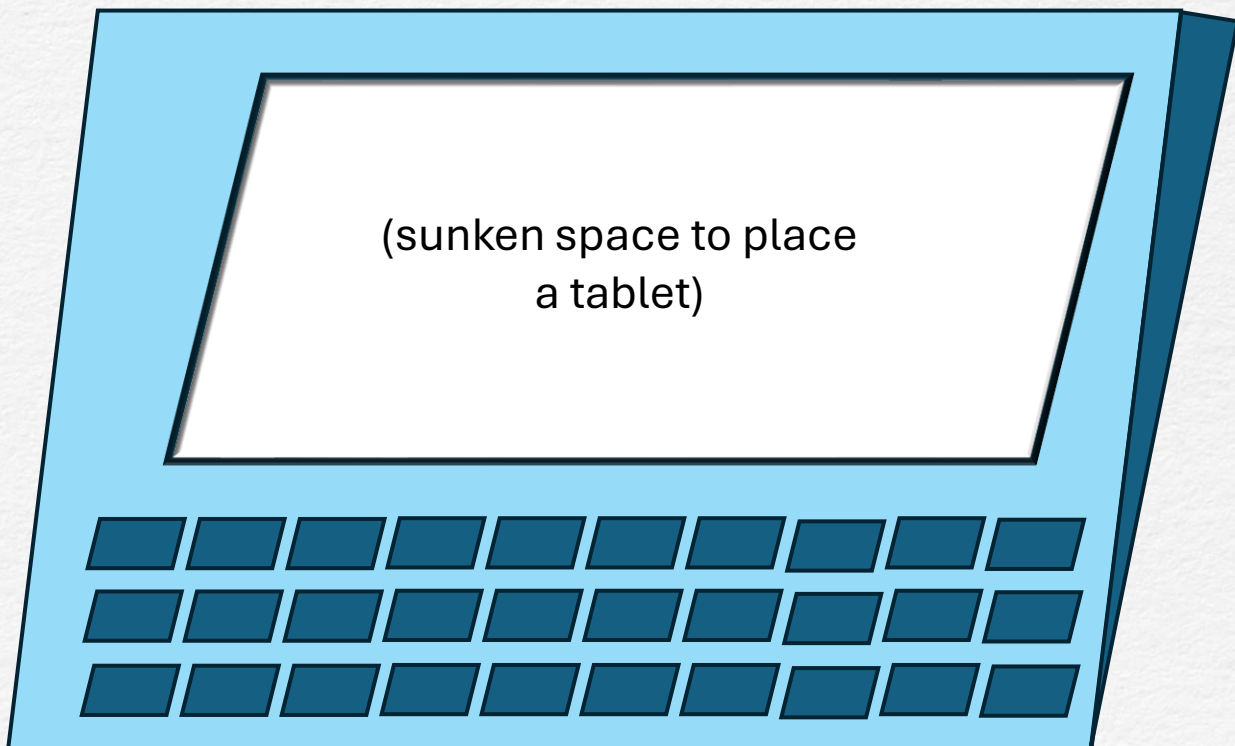


# Other designs considered

- A two-screen AIO portable device



- A tablet shell (keyboard + space to place a tablet)



# Software considerations for the Computer Technical Platform

Inside the “Coral Micro” computer there is a low-powered Linux-capable SBC that needs to boot to the coding environment in under 3 seconds.

The following environments and stacks were evaluated for implementing the coding environment.

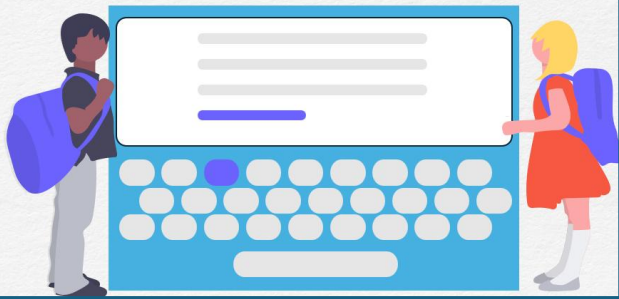
1. Dart + Flutter + Skia (no X)
2. C# + Avalonia + Skia (no X)
3. Kotlin + Compose for Desktop + Skia (no X)
4. Pascal + Ultibo (no Linux)
5. C + LVGL (no X)
6. Vala + GTK (Linux w/ X)
7. HTML5 (full screen Chromium)



Physical Product Design:

## A Dedicated Coding Computer for Young Learners

original design by Marian Veteanu



You just finished reading the paper  
“**A Dedicated Coding Computer  
for Young Learners**”.

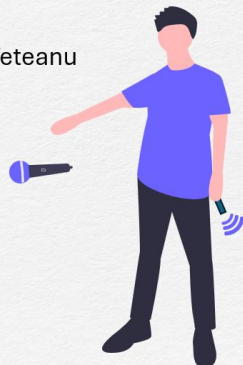
Please make sure to read my other  
two physical product design  
papers from the same series.

If you’re interested in these  
designs, or have additional  
questions, please feel free to  
contact me.

Physical Product Design:

## A Unified Clicker for Stand-Up Webinars

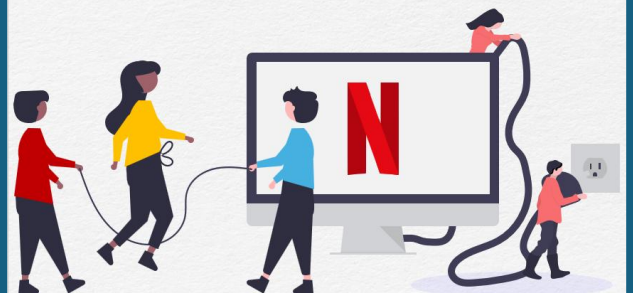
original design by Marian Veteanu



Physical Product Design:

## AI Solutions for Managing Kids' TV Time

original designs by Marian Veteanu





# Marian Veteanu

Technology Architect and Product Leader

Excited to join an organization  
where I can make an impact!

Let's connect and explore opportunities—  
message me!

<https://www.linkedin.com/in/mveteanu/>  
<https://x.com/mveteanu>

